Motivation, Requirements, and Issues for Large Scale Science Grids

William E. Johnston

Lawrence Berkeley National Laboratory and NASA Ames Research Center

wejohnston@lbl.gov

Outline

1.0 *Application Classes Motivating Widely Distributed Computing Environments*

2.0 *Vision for Science Grids*

3.0 *What are Grids?*

4.0 *What Grids Will and Will Not Do*

5.0 *Expected Outcomes*

6.0 *NASA's Information Power Grid*

7.0 *Issues for Scalability*

1.0 Application Classes Motivating Widely Distributed Computing Environments

- *Computational modeling, multi-disciplinary simulation, and scientific data analysis with a world-wide scope of participants* **– e.g. aviation safety, observational cosmology, High Energy Physics data analysis, climate modeling**

- *Real-time data analysis and collaboration involving on-line instruments*, **especially those that are unique national resources – e.g. wind tunnels, turbomachine test cells, Mars sample laboratory, LBNL's and ANL's synchrotron light sources**

- *Generation, management, and use of very large, complex data archives* **that are shared across global science communities – e.g. Earth environment data (EOS), human genome data**

- *Collaborative, interactive* **analysis and visualization of massive datasets by multi-Center teams – e.g. wind tunnel data, air/space frame design data, DOE's Combustion Corridor project**

Addressing the requirements of these classes of applications in a general way, with common Grid infrastructure deployed across the DOE Labs, NASA Centers, and collaborating universities, will enable many different large-scale applications to routinely use widely distributed resources.

Multi-disciplinary Simulations

Multi-disciplinary simulations provide a good *example of a class of applications that are very likely to require aggregation of distributed computing, data, and intellectual resources*.

Such simulations – e.g. whole system aircraft simulation and whole system living cell simulation – *require integrating applications and data that are developed by multi-disciplinary teams of researchers* who are frequently in different locations.

The research teams are typically the only ones that have the expertise to maintain and improve the simulation code and/or the body of experimental data that drives the simulations. This results in an inherently distributed computing and data management environment.

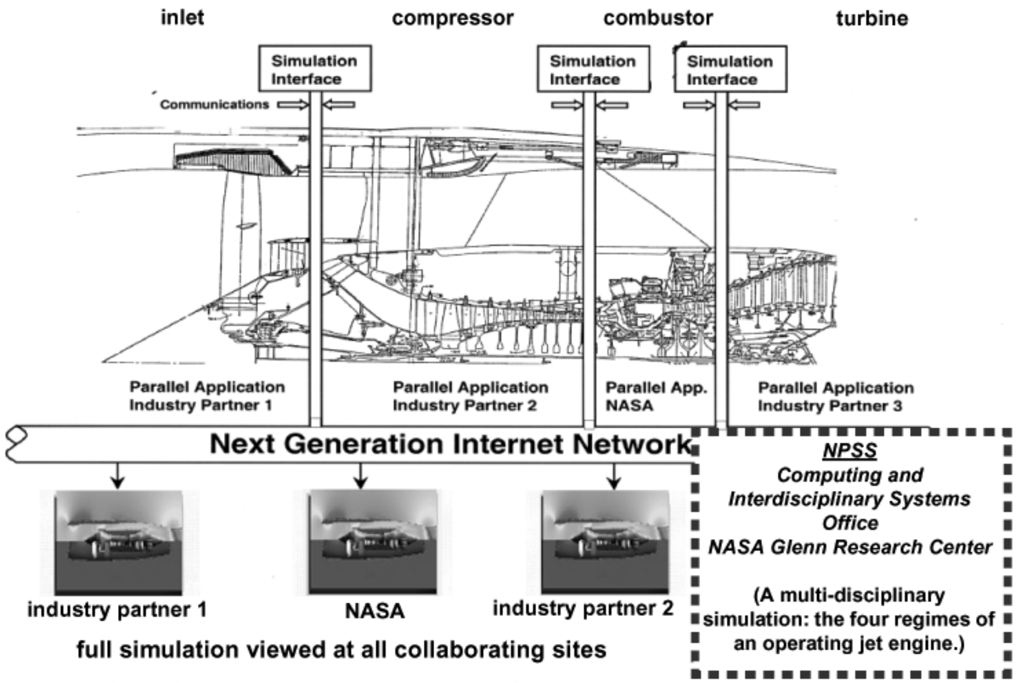
Consider a vision for Aviation Safety:

How do we simulate the entire commercial airspace of the country?

(Yuri Gawdiak (VNAS) and Bill McDermott, NASA Ames, John Lytle and Gregory Follen, NASA Glenn (NPSS)).

This vision is being approached through a set of increasingly complex and computationally intensive integrations:

Component simulations are combined to get a system simulation.



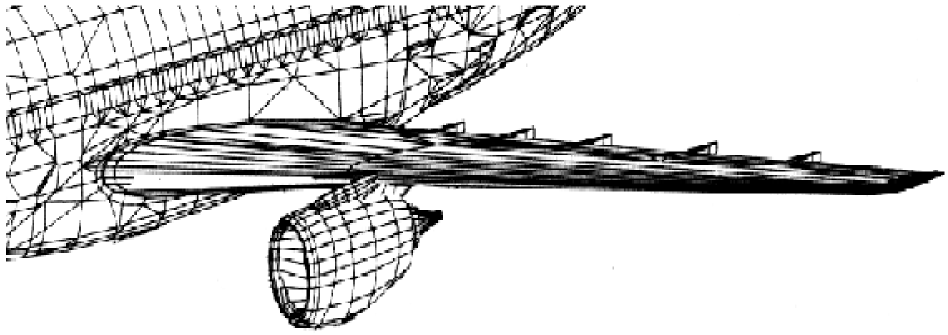
Information Technology

Jet Engine System

Issues for combining component simulations

- wrapping the simulation code
- composing these codes
- coordinating resources for executing the multiple components

Multiple system simulations are coupled to represent pieces of a device.



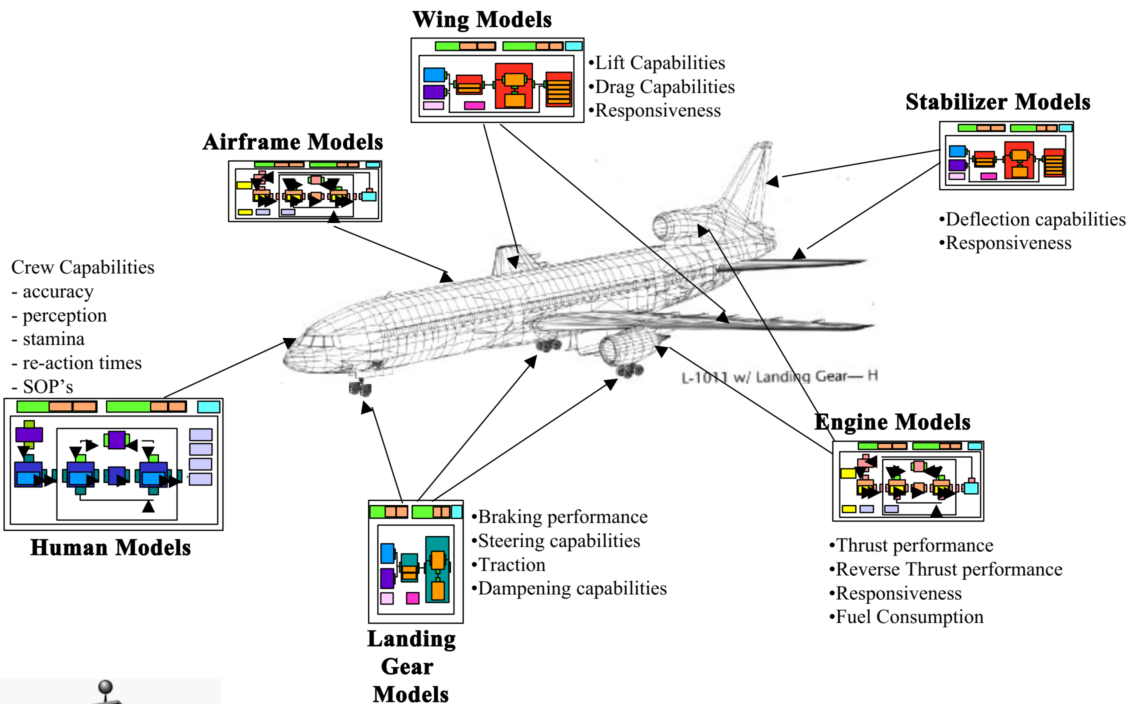
Engine System + Wing System

Motivating Applications - Multi-Disciplinary Simulations

Multi system simulation issues:

- multi-Center interactions - component parameters maintained by discipline experts
- shared compute and data resources

Whole device simulations are produced by coupling all of the subordinate system simulations.



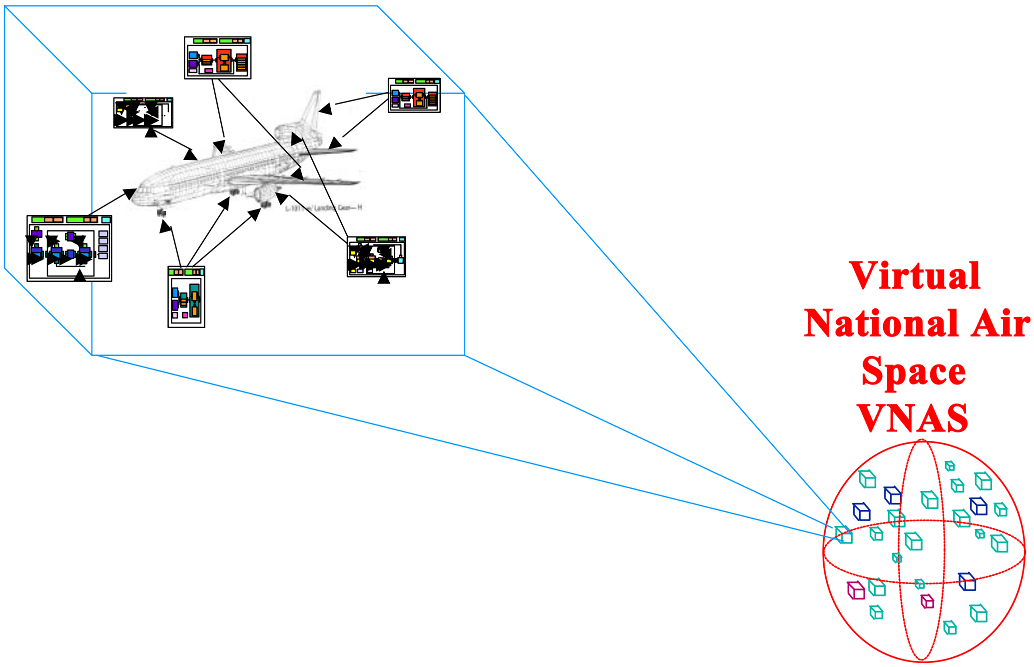
Information Technology

Motivating Applications - Multi-Disciplinary Simulations

Whole device simulation issues:

- increasingly complex interaction of models and data
- scaling of computing and networking capacity by O(10)

Devices are inserted into a realistic environment.



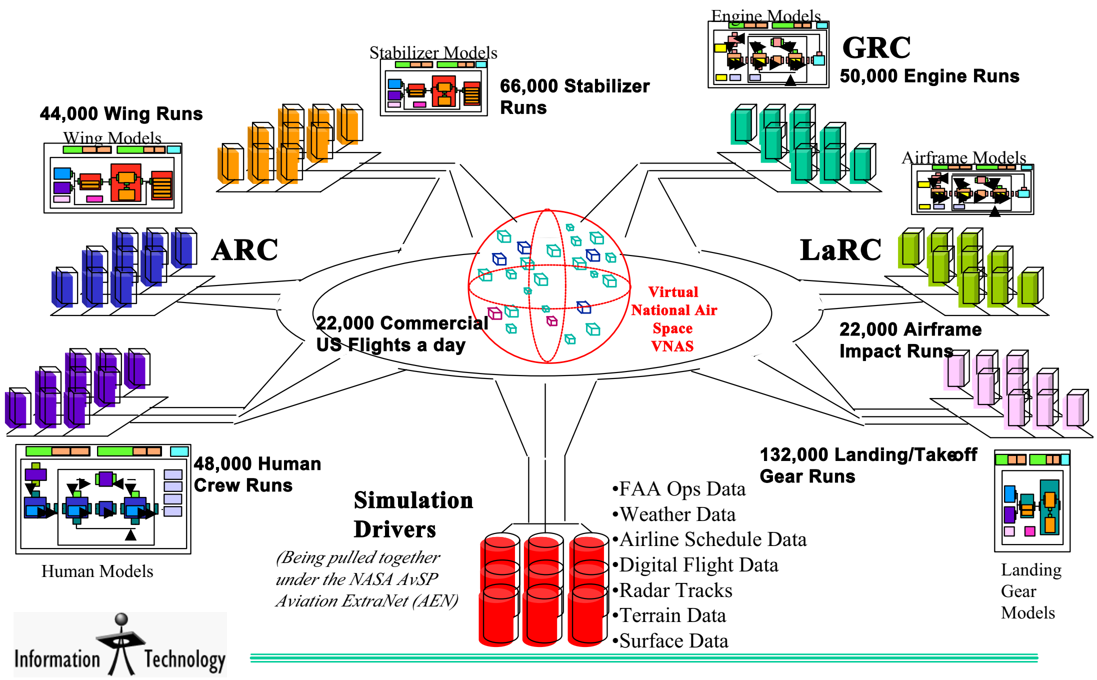
Information Technology

Motivating Applications - Multi-Disciplinary Simulations

Issues:

- the operating environment must be incorporated into the simulation, which effectively established further coupling of the system simulations

Devices and environment are combined for operational systems simulation.



Motivating Applications - Multi-Disciplinary Simulations

Issues:

- overall system operational models and parameters must be incorporated, which introduces “indirect device” interactions
- scale computing and networking by O(10000)

Clearly such simulations will need to use aggregated computing, data, instrument, and intellectual resources across multiple NASA Centers.

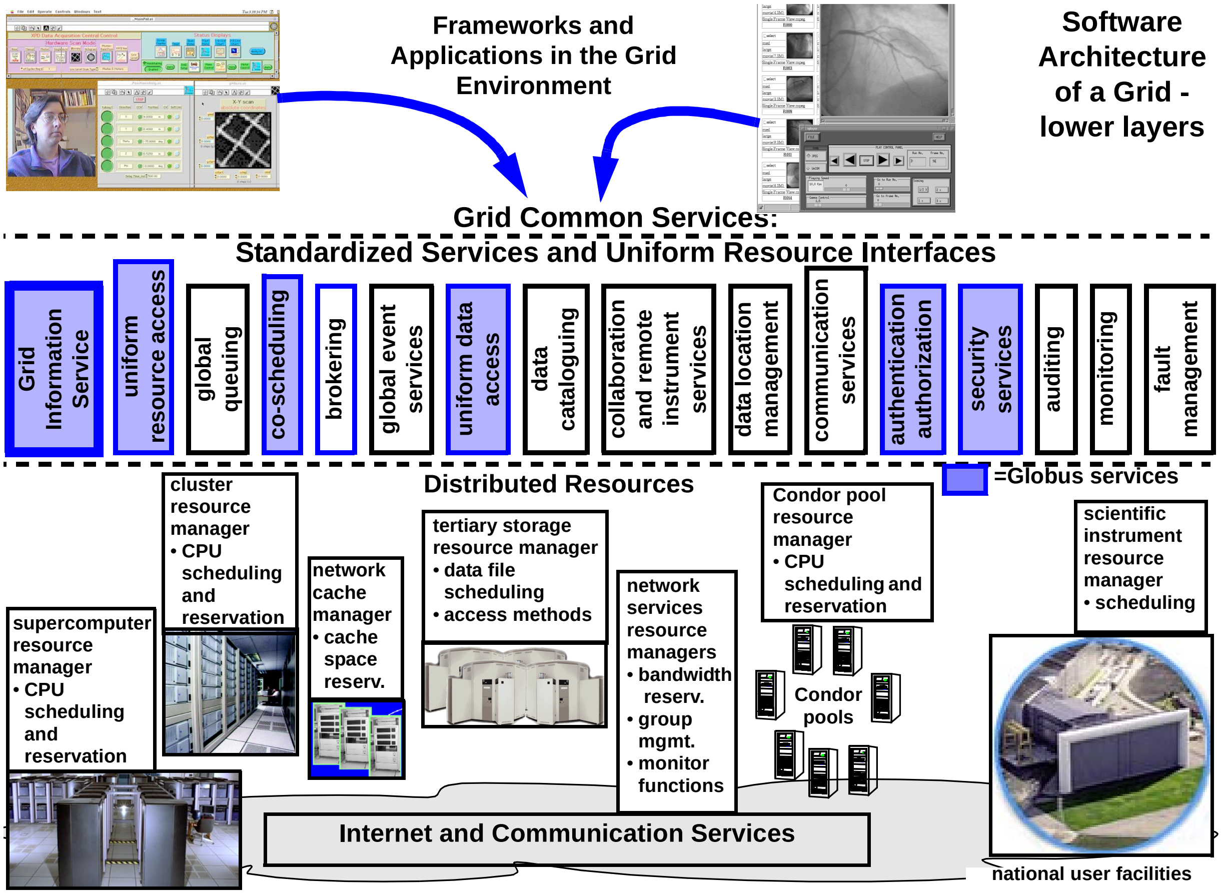
2.0 Vision for Science Grids

The vision for “Grids” is to revolutionize the use of computing in science and engineering. This will be accomplished by making the construction and use of large scale systems of diverse resources as easy as using today’s desktop environments. This will enable the degree of scalability in scientific and engineering computing necessary for NASA and DOE to address very large simulation and data analysis problems.

3.0 What are Grids?

The type of Grid being described here is based on services that are defined by their protocols and interfaces. In this context, Grids are tools, middleware, and services for

- **providing a uniform look and feel to a wide variety of computing and data resources**
- **supporting construction, management, and use of widely distributed application systems**
- **facilitating human collaboration and remote access and operation of scientific and engineering instrumentation systems**
- **managing and securing the computing and data infrastructure**



Software Architecture of a Grid - upper layers

Problem Solving Environments

- Tools to implement the human interfaces
- Mechanisms to express, organize, and manage the workflow of a problem solution
- Access control
- E.g. SciRun [19], Ecce [20], "portals", WebFlow [21],...

Applications and Supporting Tools

application codes | visualization toolkits | collaboration toolkits | instrument management toolkits | data publication, subscription, and access toolkits

Grid enabled libraries

Application Development and Execution Support Services and Systems

Globus MPI | CORBA | Condor-G | Legion | Java/Jini | OLE/DCOM

Grid Common Services

Distributed Resources

3.1 Notes on Grid Architecture

- **Problem Solving Environments are the user interface to Grids, and are supported by Grid toolkits for**
  - **job submission, control, and tracking services**
  - **workflow management for specific classes of applications (e.g. physics data analysis frameworks or aircraft design parameter study managers)**
  - **policy based access control, etc.**

- **Application development tools and services support various styles of programming in the Grid environment, as well as the development of Grid services themselves. E.g.:**
  - **uniform data access methods developed in the DataGrid project [13] will form the foundation for global storage management services such as MCAT/SRB [17] and the Storage Access Coordination System (STACS), HRM (HPSS Resource Manager) [14]**
  - **Globus I/O enabled MPI library provides coordinated, MPI communication between processes on separate systems**

- **The “Grid Common Services” locate, schedule, and provide uniform views of the underlying resources. E.g.: resource access, naming and location, and co-scheduling for computing, networking, and instrument systems**

- **Resource managers provide the basic functionality and access for the actual resources**
  - **some already exist - e.g. batch schedulers - however do not always support the required functionality (e.g. for advance reservation)**

4.0 What Grids Will and Will Not Do

- **Grids provide common resource access technology and operational services deployed across virtual organizations. This allows the possibility of sharing resources, but does automatically permit it:**
  - **local <u>authorization models are not changed</u> by the Grid.**
  - **common Grid technology will allow standardized views of resources and uniform access to resources, thereby permitting very large application systems to be built, and <u>if policy permits</u>, to share resources across sites and organizations.**

- **Grids <u>will enable large scale applications based on</u>:**
  - **Loosely coupled computations: Simulation parameter sweeps and certain types of experiment data analysis involve initiating and managing 100s, 1000s, and 10000s of processes. Grids provide the access and mechanisms for using large numbers of computing and data resources for this type of calculation.**
  - **Large scale pipelined applications: Multi-component simulations involve executing multiple, coupled, medium to large scale simulations on multiple computing resources. Grids provide co-scheduling and data stream management to support this.**

- **Grids <u>will not, in the near term</u>, enable very large, single problems such as CFD calculations to be spread across distributed systems.**
  - **To accomplish this we will need new approaches and algorithms that are tolerant of high and variable latency. There is R&D going on to address this issue in the long term.**
- **Grids <u>will not provide</u> a lot of “free” resources.**
  - **To produce a highly capable science Grid organizations must place major resources on the Grid.**

5.0 Expected Outcomes

Grids will provide a *uniform usage and management interface* to computing, collaboration, storage, and instrument systems, and together with the *capability of dynamically and scalably connecting* these into large, on-demand systems. This should lead to:

- **Increased mobility of human expertise and increased access to computing and data by computational scientists**
- **Routine collaboration among NASA Centers and DOE Labs, and their university partners through ready and secure access to collaboration tools, remote instruments, and petabyte size data sets**

Expected Outcomes

- **Easily used, application oriented, user interfaces (problem solving environments / workbenches) that provide access to powerful, diverse, and widely distributed resources**
- **The ability to build large-scale problem solving systems that are built dynamically from aggregated resources will support multi-disciplinary scientific and engineering computing and data based activities that are not steady state - i.e. those that may require a different resource mix for every different problem**

- **New approaches to laboratory science through the coupling of large-scale computing and storage systems to instrument systems in order to provide real-time analysis of experiment data and feedback based experiment control**

- **Standardized services and tools that make it easier to incorporate new computer architectures, data systems, and instruments into a usable application environment**

- **A pool of resources that has standardized capabilities, aggregation strategies, and management so that large-scale systems could be quickly built for emergency response situations**

6.0 NASA's Information Power Grid

IPG [5] is NASA's project to build a functioning Grid that manages and provides access to the supercomputing facilities at the Numerical Aerospace Simulation Systems Division of NASA Ames Research Center.

6.1 Approach and Goals for NASA's IPG

- Grids are built through collaborative efforts, and at the same time facilitate collaboration: IPG is a collaboration among several NASA Centers and the NSF Supercomputer Center PACI consortia, with the Grid Forum [11] providing "coordination" of many institutions world wide
- Deployment of existing technology (Globus [1], Condor [18], Grid portals [10], etc.) will provide for relatively rapid impact – the NASA IT/ANCS [7] program is providing computing and storage resources for a prototype production IPG environment

Approach and Goals

- The Ames NAS division [6] will provide the development and support for critical Grid services – this will ensure persistent and usable infrastructure across the NASA Centers
- Grid support for building collaboration services will both facilitate construction of the Grid and more readily provide collaboration tools to users
- The IPG operational model provides for easy user access across Centers and for local control of resources that are connected to IPG

- Strong security will be provided from the start in order to address authentication, authorization, and infrastructure assurance in open science networks for both applications and Grid services
- As Grid services are debugged and validated they will be offered to NASA's production supercomputing organization (CoSMO [8]) as a means of providing a uniform supercomputing environment – the production IPG represents a new service delivery model for NASA computing, data, and instrument resources

6.2 IPG Milestones Met to Date

IPG Milestone 1: High Speed Distributed Data Access

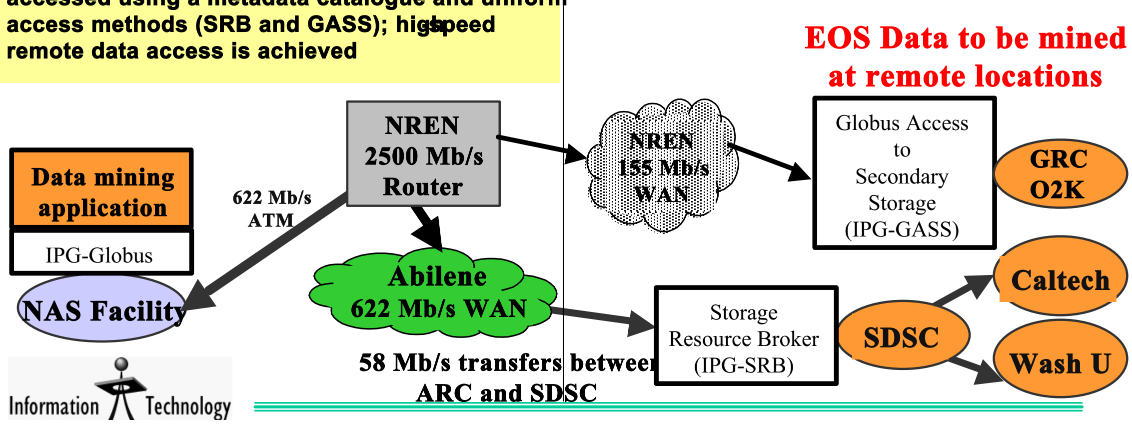
Background A key function of Grids is to provide uniform access to widely distributed resources, including heterogeneous distributed archival data and information systems

Objective Demonstrate software to enable seamless access to catalogued archival data and information that is distributed throughout multiple NASA Centers and collaborator sites

Accomplishment Multiple data archive sites are accessed using a metadata catalogue and uniform access methods (SRB and GASS); high-speed remote data access is achieved

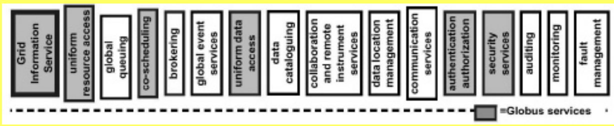
Significance On-demand access to widely distributed archived data and information; enhanced engineering and scientific collaboration

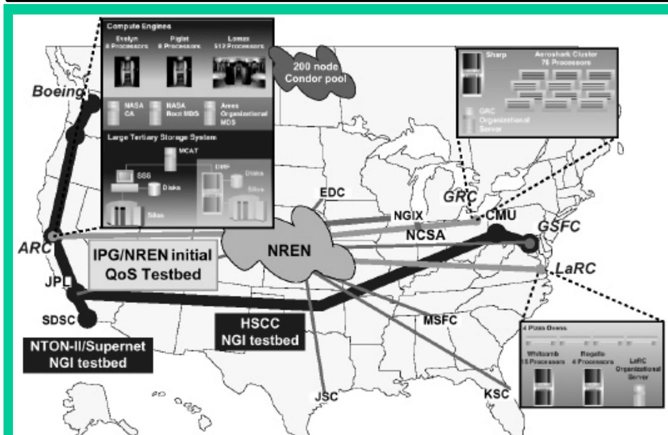
Future Plans Make this capability a permanent part of the IPG infrastructure and integrate with archival storage systems at ARC, GSFC, and JPL; incorporate the IPG/Globus security mechanisms to provide strong access control and secure remote data access



IPG Milestone 2 Heterogeneous Computing

IPG Grid Common Services: Standardized services and uniform resource access



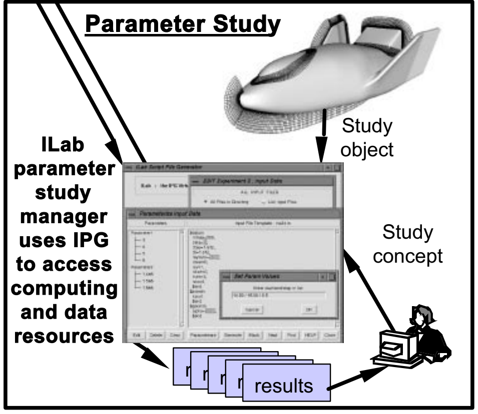


IPG managed compute and data management resources

Condor Cycle Scavenging

- Application is molecular design for nanotechnology devices and materials
- Uses 0.5 million CPU hours/year scavenged from a 60-100 Sun and SGI workstations - a subset of the NAS Condor pool
- Application is coded in Java for platform independence
- The Condor system is an IPG middleware service

Parameter Study



Information Technology

IPG Milestone 2 - 4QFY00

Milestone: A prototype, heterogeneous, distributed computing environment.

IPG is a "Grid," and as such provides the middleware services for building large-scale, dynamically constructed problem solving environments from distributed, heterogeneous resources.

Metric: System tools and software provided; testbeds (2 or more classes of machines) at 3 NASA centers linked; application demonstration completed.

IPG software services provide for resource discovery, uniform access to geographically and organizationally dispersed computing and data resources, job management, single sign-on, security, inter-process communication, and resource management.

Both the services and their operational support are in place at ARC, GRC, and LaRC.

Hardware resources for the baseline IPG prototype-production system include approximately 600 CPU nodes in half a dozen SGI Origin 2000s distributed across the three NASA centers, 100 Terabytes of uniformly and securely accessible mass storage, several workstation clusters involving about 100 CPUs, and Condor pool of 200 workstations.

Outcome: Reduction in end-to-end turnaround time for aerospace simulation problems; peak performance, cost performance

The ILab parameter study system provides a substantial human efficiency in studying complex systems, resulting in reduction of turnaround time for system simulation.

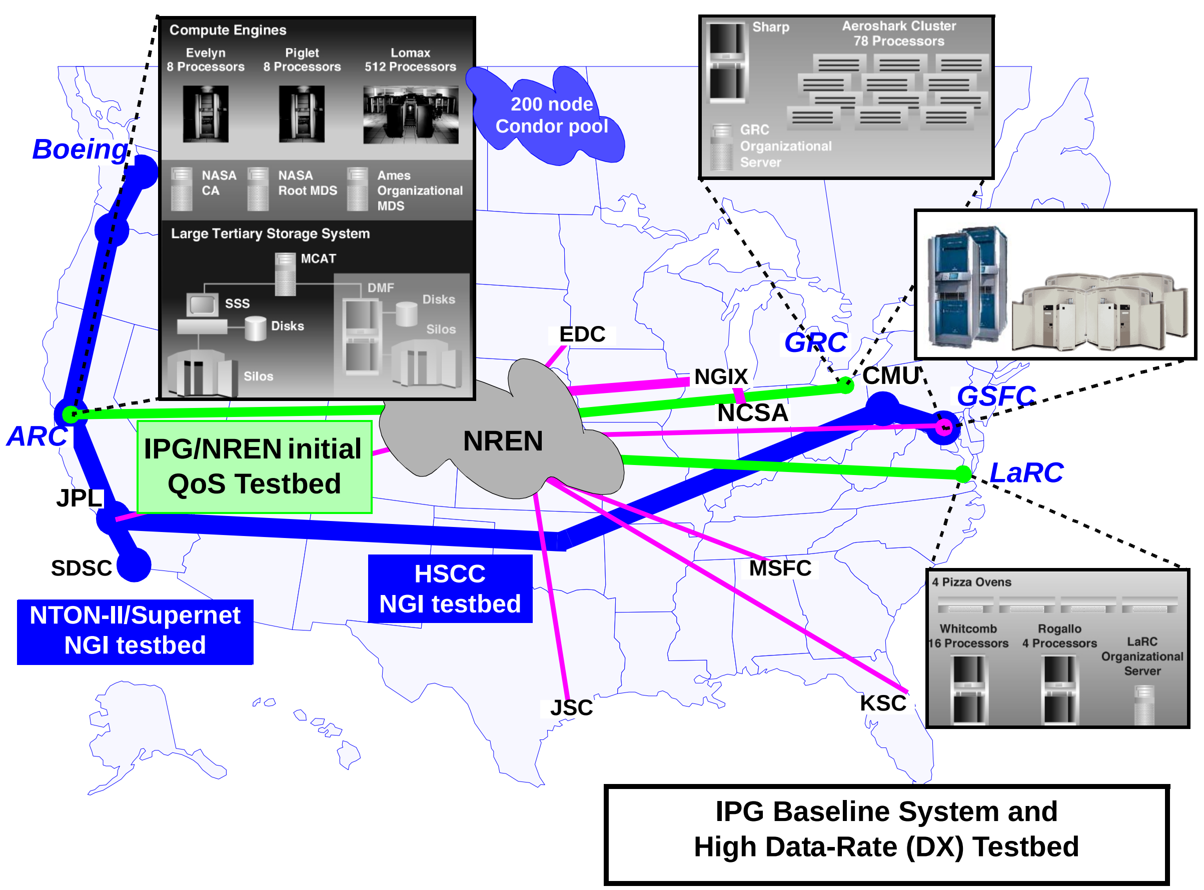
The molecular design application coded in Java and managed by the Condor cycle scavenger is able to apply hundreds of thousands of megaflop years of otherwise idle computing time to significant NASA problems.

www.ipg.nasa.gov

Information Technology

6.3 IPG Baseline Operational System - 10/2000

- Computing resources: ≈600 CPU nodes in half a dozen SGI Origin 2000s and several workstation clusters at Ames, Glenn, and Langley, with plans for incorporating Goddard and JPL, and approx. 270 workstations in a Condor pool
- Communications: High speed, wide area network testbed among the participating Centers
- Storage resources: 30-100 Terabytes of archival information/data storage *uniformly and securely* accessible from all IPG systems



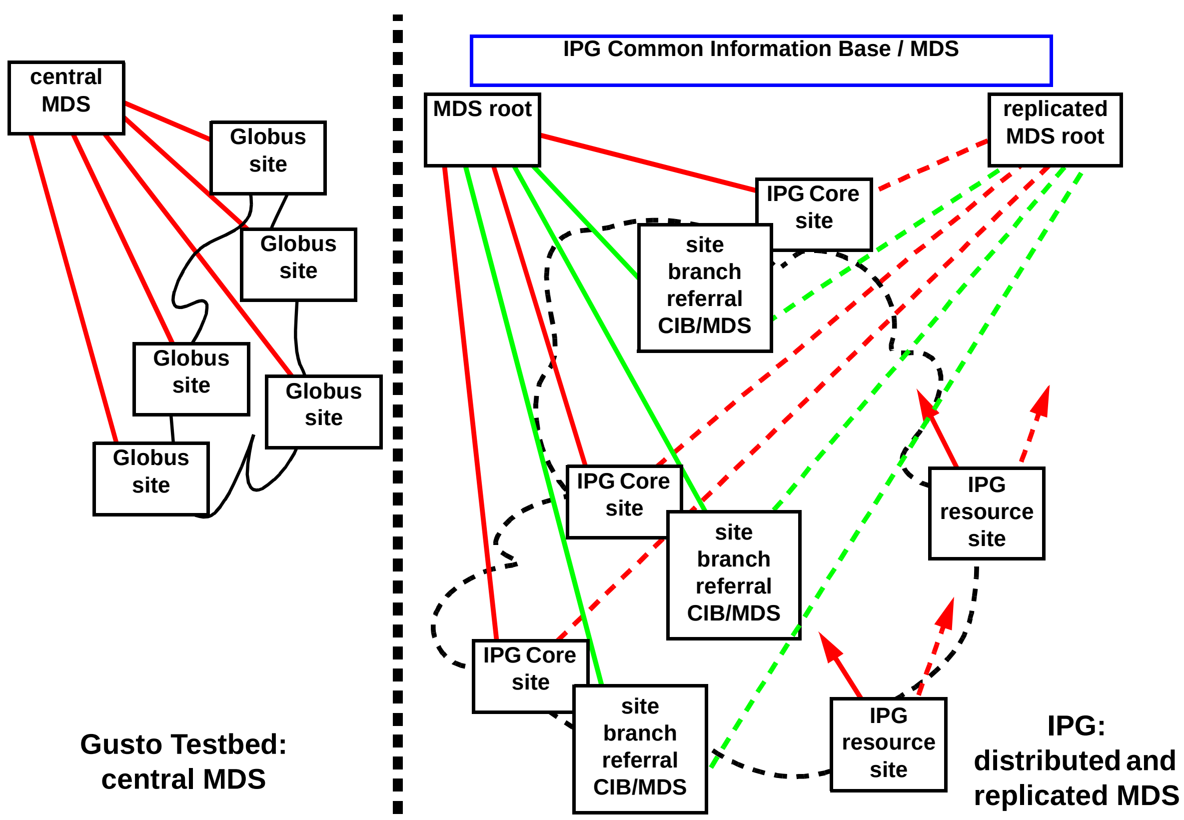
- Globus [1] providing the Grid common services
- Grid programming and program execution support
  - Grid MPI (via the Globus communications library)
  - CORBA integrated with Globus
  - global job queue management
  - high throughput job manager
  - Condor [18] (“cycle stealing” computing)
- A stable and supported operational environment
- A stable and supported user environment

- Several “benchmark” applications operating across IPG (parameter studies, multi-component simulations, multi-grid CFD code)
- Multi-Grid operation (e.g. applications operating across IPG and NCSA)

6.4 How is IPG Being Accomplished?

1) Persistent operational environment that encompasses significant resources

- **"IPG Prototype Startup Tasks (target: 3/00)**
  - **Globus deployed across Ames, GRC, and LaRC (Task 1.0)**
  - **IPG common grid information base (GIS/MDS) (Task 2.0)**
  - **Security via Globus Security Infrastructure, IPG X.509 Certification Authority and certificate server (Task 3.0)**
  - **Global queuing and user-level queue management capability on top of Globus (Task 4.0)**



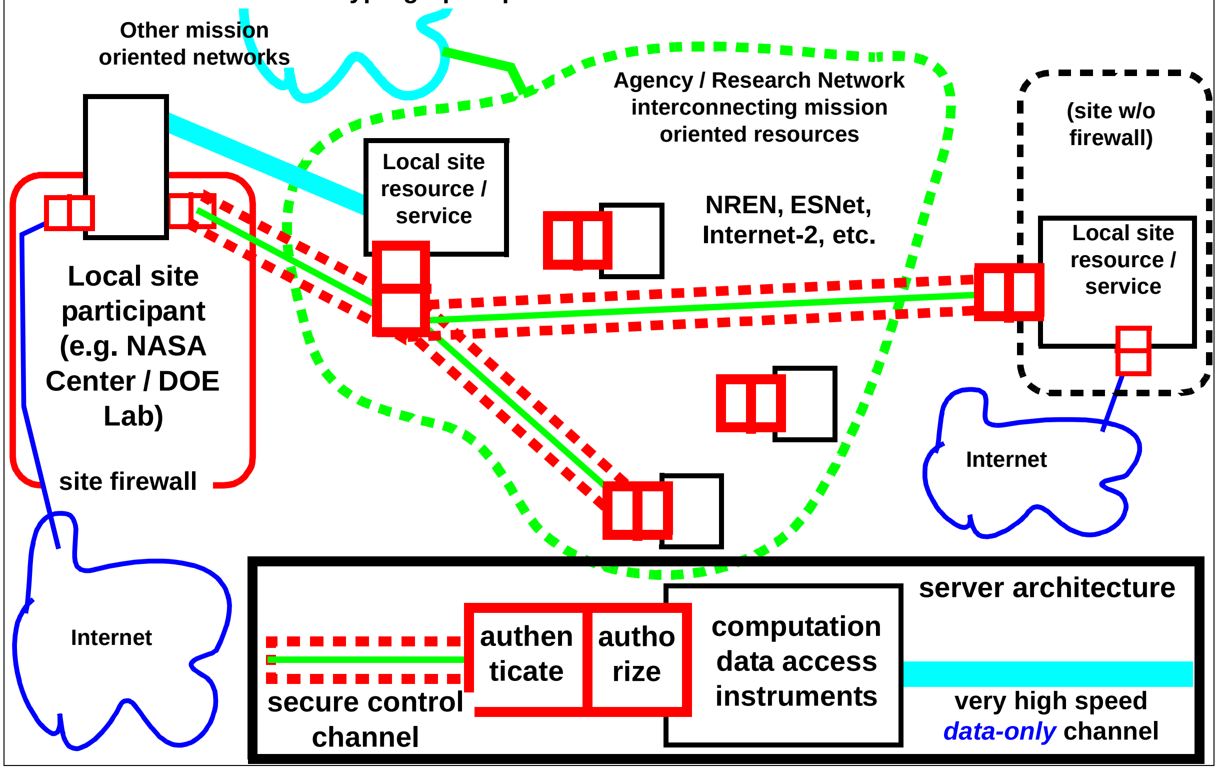
Migration of Common Information Base from R&D to Prototype Production

- Computing resources for the initial IPG multi-center testbed (CX) (Task 5.0)
- Networking for the IPG Testbed: Ames, GRC, LaRC (Task 6.0)
- IPG Access for Archival and Published Data: SDSC's Metadata Catalogue (MCAT) and the Storage Resource Broker (SRB) (Task 7.0)
- Heterogeneity in the IPG testbed: Condor (Task 8.0)
- Heterogeneity in the IPG testbed: High performance clusters (Task 9.0)

♦ **IPG Operational Tasks (needed for v.1.0 of IPG baseline - 10/2000)**

- Security model (Task 10.0)
- IPG Information Base / MDS database maintenance (Task 11.0)
- IPG/Globus system administration (Task 12.0)
- Automatic monitoring of IPG components (Task 13.0)
- Trouble ticket model (Task 14.0)
- Condor support (Task 15.0)
- CORBA support (Task 16.0)
- Documentation (Task 18.0)

Security Model: All command and control functions are transported over an encrypted channel, after the client/user is authenticated and authorized. This compartmentalizes all servers: If multiple servers are involved in a distributed system, then each reauthorizes connections through the use of cryptographic proxies or active re-authentication.



Accomplishing the Baseline - Operational Environment

- **User services (Task 19.0)**
- **Account management (automated generation and maintenance mechanisms) (Task 20.0)**
- **Globus with multiple MDS and PKI CAs (Task 21.0)**
- **Allocation Management and Accounting (Task 22.0)**
- **System testing: Verification suites, benchmarks, and reliability/sensitivity analysis for IPG (both static and dynamic) (Task 23.0)**

- **IPG Functionality Tasks (near-term - 10/2000)**
  - **CORBA in the IPG environment. (Task 24.0)**
  - **Integration of Legion (Task 25.0)**
  - **CPU resource reservation (Task 26.0)**
  - **High Throughput Computing (Task 27.0)**
  - **Programming Services (Task 28.0)**
  - **Distributed debugging (Task 28.1)**
  - **Grid enabled visualization (Task 28.2)**

- **IPG Functionality Tasks (mid-term - 10/2001)**
  - **Network bandwidth reservation (Task 29.0)**
- **Characteristic Applications - 10/2000**
  - **OVERFLOW port & tune (Task 31.0)**
  - **NPSS [23] port & tune (Task 32.0)**
  - **Parameter study**
  - **Heterogeneous testbed application: Condor (Task 34.0)**
  - **Heterogeneous testbed application: High performance clusters (Task 35.0)**

7.0 Issues for Scalability

Deployment of sizable, prototype Grids will reveal many issues for scalability, and R&D will have to address those issues.

Two areas that have been revealed so far as discussed here: Grid Information Services, and distributed authorization.

7.1 Grid Information Services

The Grid will be a global infrastructure, and it will depend heavily on the ability to locate information about computing, data, and human resources for particular purposes, and within particular contexts.

Most Grids will serve *virtual organizations* whose members are affiliated by a common administrative parent (e.g. the DOE Science Grid and NASA's Information Power Grid), common long-lived project (e.g. the High Energy Physics, Atlas experiment), common funding source, etc.

In this paper we present user/functional requirements and operational requirements, followed by an

implementation proposal and it’s issues.

The question of using a hierarchical structure vs., e.g., very fast searches on flat attribute spaces (like Web search engines), is an open issue. However, it will be clear from some of the nomenclature and assumptions in the requirements section that a hierarchical structure is somewhat implicit in the discussion. This issue will be revisited at the end of the requirements section.

7.1.1 User Requirements

7.1.1.1 Searching

The basic sort of question that a GIS must be able to

answer is for all resources in a virtual organization, provide a list of those with specific characteristics.

For example:

> “Within the scope of the Atlas collaboration, return a list of all Sun systems with at least 2 CPUs and 1 gigabyte of memory, and that are running Solaris 2.6 or Solaris 2.7.”

Answering this question involves filtering on both the virtual org. attribute and the resource attributes in order to produce a list of candidates.

7.1.1.2 Virtual Organizations

It should be possible to provide “roots” for virtual organizations. These nodes provide search scoping by establishing roots that sit at the top of a hierarchy of virtual org. resources, and therefore starting places for searches. Like other named objects in the Grid, these virtual org. nodes might have characteristics specified by attributes and values. In particular, the virtual organization node probably needs a name reflecting the org. name, however some names (e.g. for resources) may be inherited from the Internet DNS domain names

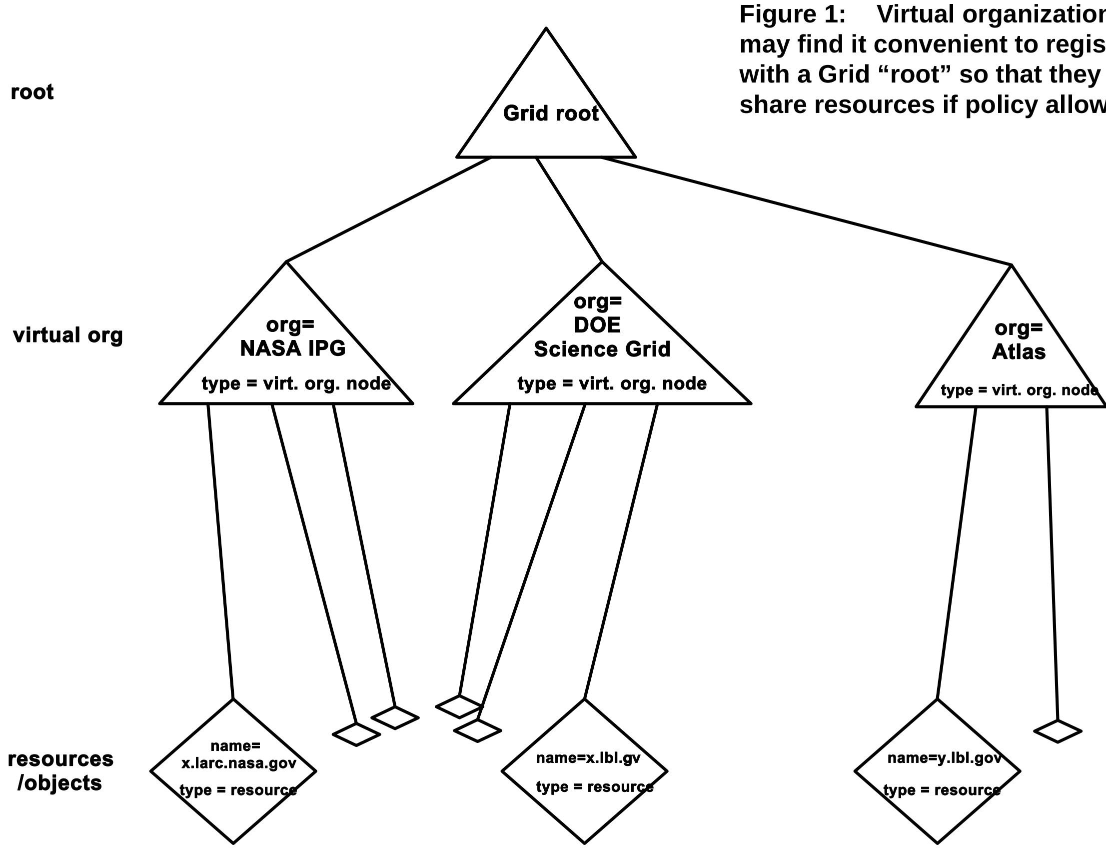


Figure 1: Virtual organizations may find it convenient to register with a Grid “root” so that they can share resources if policy allows.

7.1.1.3 Information and Data Objects

A variety of other information will probably require cataloguing and global access, and the GIS should accommodate this in order to minimize the number of long-lived servers that have to be managed:

- **dataset metadata**
- **dataset replica information**
- **database registries**
- **Grid system and state monitoring objects**
- **Grid entity certification/registration authorities (e.g. X.509 Certificate Authorities)**
- **Grid Information Services object schema**

Therefore it should be possible to create arbitrary nodes to represent other types of information, such as information object hierarchies.

This sort of information has to be consistently named in a global context, will have to be locatable, and in some cases will have an inherently hierarchical structure.

Requirements for these catalogues include:

- **providing unique and consistent object naming**
- **access control**
- **searching, discovery, and publish/subscribe**

7.1.2 Operational Requirements

7.1.2.1 Performance and Reliability

- Queries, especially local queries, should be satisfied in times that are comparable to other queries like uncached DNS data. E.g., seconds or fractions of seconds.
- Local sites should not be dependent on remote servers to locate and search local resources.
- It should be possible to restrict searches to local resources of a single, local, administrative domain.

- Site administrative domains may wish to restrict access to local information, and therefore will want control over a local, or set of local, information servers.

These imply the need for servers intermediate between local resources and the virtual org. root that are under local control for security, performance management, and reliability management.

(Note that in the Globus terminology that these intermediate directory servers are called *GIIS*s.)

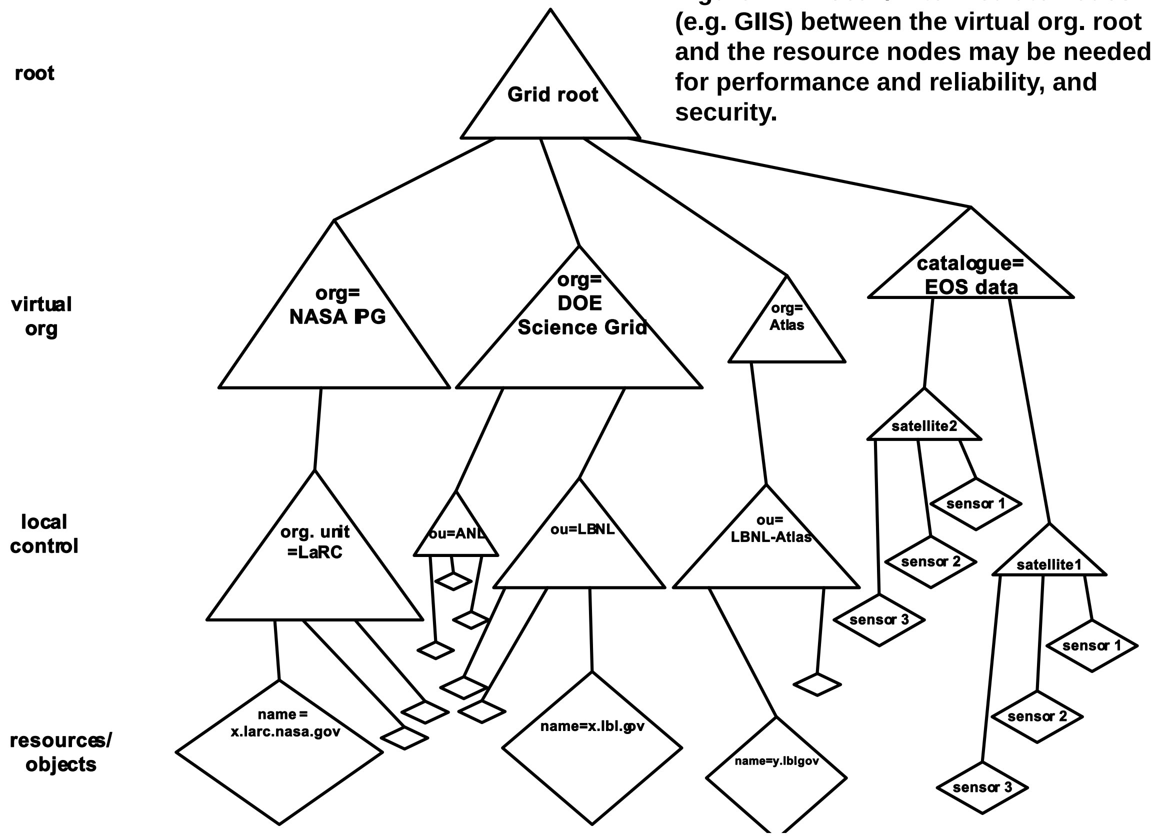


Figure 2: Local / intermediate nodes (e.g. GIIS) between the virtual org. root and the resource nodes may be needed for performance and reliability, and security.

7.1.2.2 Multiple Membership

Many objects/resources will have membership in multiple virtual organizations. This information, like other resource attributes, will likely be maintained at the resources in order to minimize management tasks at the upper level nodes.

- **It must be possible for a resource to register with multiple virtual organizations.**

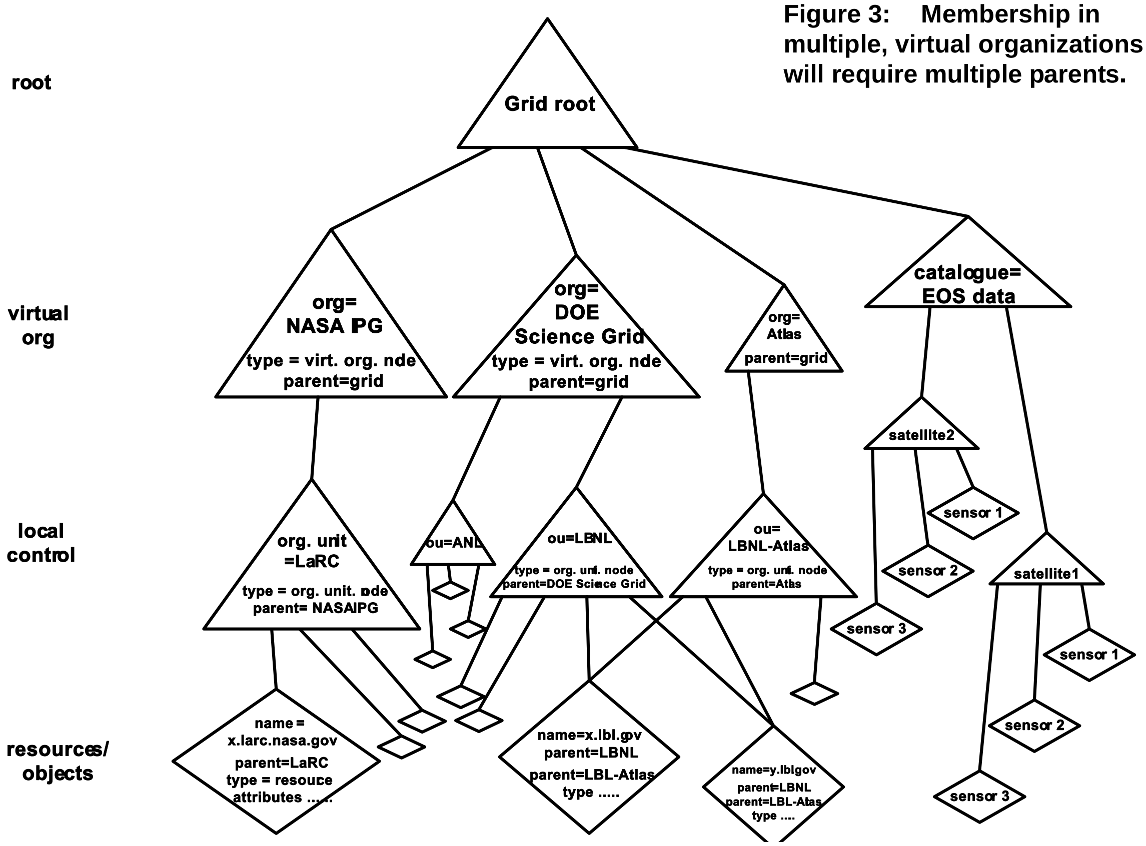


Figure 3: Membership in multiple, virtual organizations will require multiple parents.

7.1.3 Operational Requirements

7.1.3.1 Minimal Manual Management

The management of the information servers above the resources (in the case of a resource catalogue) must be as automatic/minimal as possible.

- **Information about a resource should be maintained at that resource, and should propagate automatically to superior information servers.**

7.1.3.2 Control over Information Propagation

At each level of information management (four have emerged so far) there are various reasons why both import and export controls will have to be established.

- **At the object / resource level (see Figure 4), the local administrators must have control over what information is exported for the purposes of registration.**
- **At the object / resource level there must be access control mechanisms to restrict the types of queries or the detail that queries return.**
- **The nodes at the level of "local control" are meant to model a common system administration domain, and must support a common security policy, including who is allowed to register (import control) and what information is passed outside of the**

security domain (export control). It should, e.g., be possible to implement policies such as making anonymous the information that is passed to the next level up (either for registration or as search results).

Such anonymous information should allow broad searches at the upper levels, but limit specific searches to the lower levels, where searches can be authorized based on the relationship of the searcher to the resource.

- The same sorts of capabilities as exist at the local control level must be available at the virtual organization level in order to maintain control over the characteristics of the virtual organization.
- At the root, again it must be possible to apply policy to registration (e.g. to prevent nodes below the virtual org. level from registering at the root).
- The ability to do automatic node replication for reliability will exist at all levels.

7.1.3.3 Performance and Robustness

Finally, when we look at the information flow paths (Figure 5), it is apparent that a lot of information may

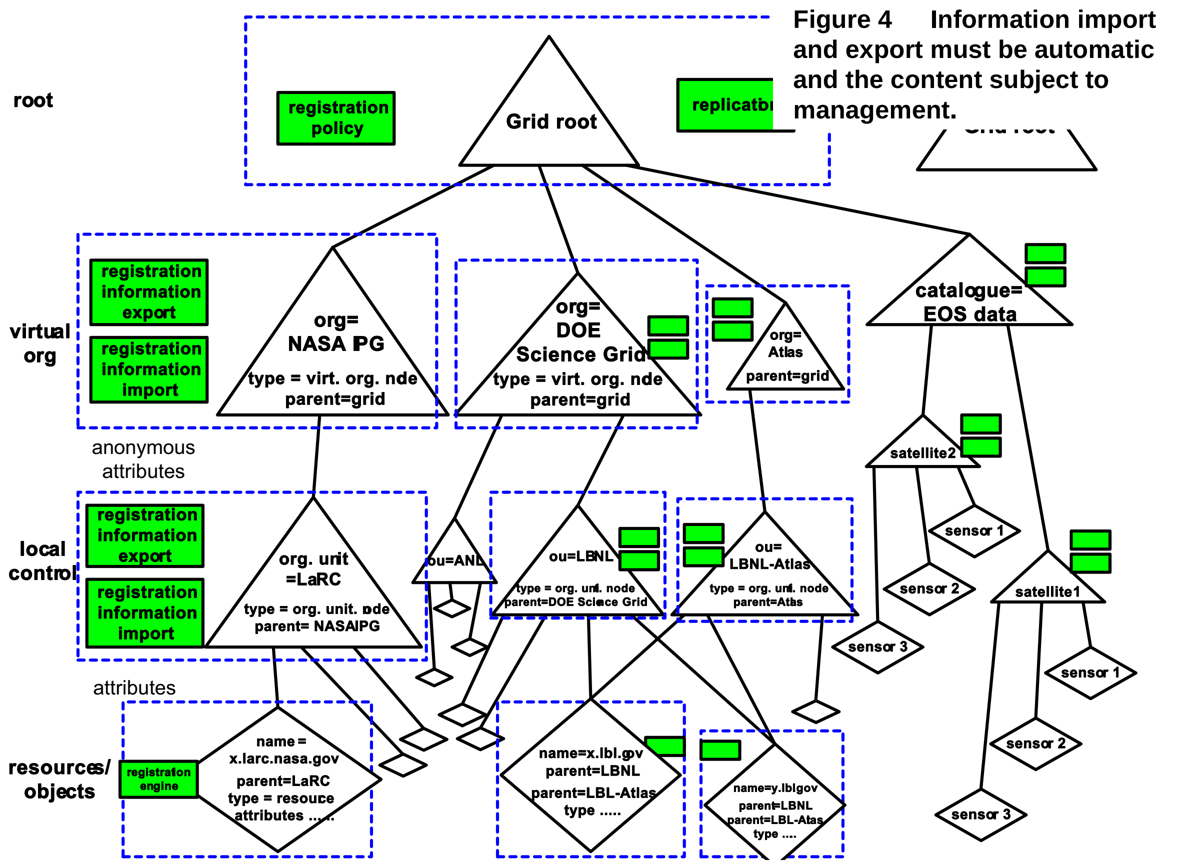
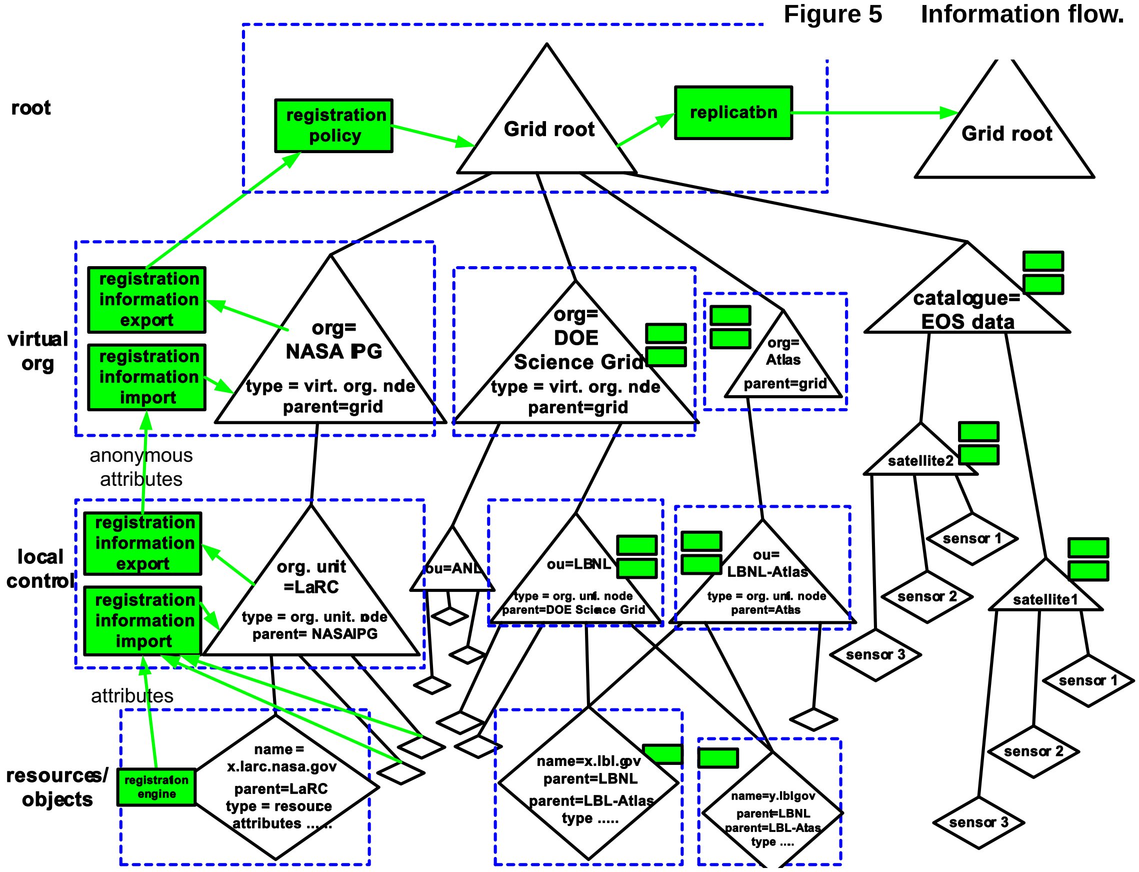


Figure 4 Information import and export must be automatic and the content subject to management.

flow in complex patterns. Well tested components and procedures will be needed. We will probably need some modeling or measurement information on the volume and rate of data flow in such an environment in order to assess the scaling issues.

Figure 5 Information flow.



7.2 Access Control for Widely Distributed Systems

As we gain experience with security in geographically and organizationally diverse environments, we find that while it is possible to come up with a model and an implementation that deals with the various issues, the deploying and using such models and implementations presents a myriad of scaling problems: user education and acceptance, credential and key management, cross-domain naming issues, etc.

Here we describe the Akenti system [15] that was designed and deployed in DOE's distributed collaboratory environment, and, we believe, provides a fairly representative example of the issues for distributed authorization for Grids.

7.2.1 Motivation

Our scientific environment involves

- **multi-user instruments at national facilities**
- **widely distributed supercomputers and large-scale storage systems**
- **data sharing in restricted collaborations**
- **network-based multimedia collaboration channels**

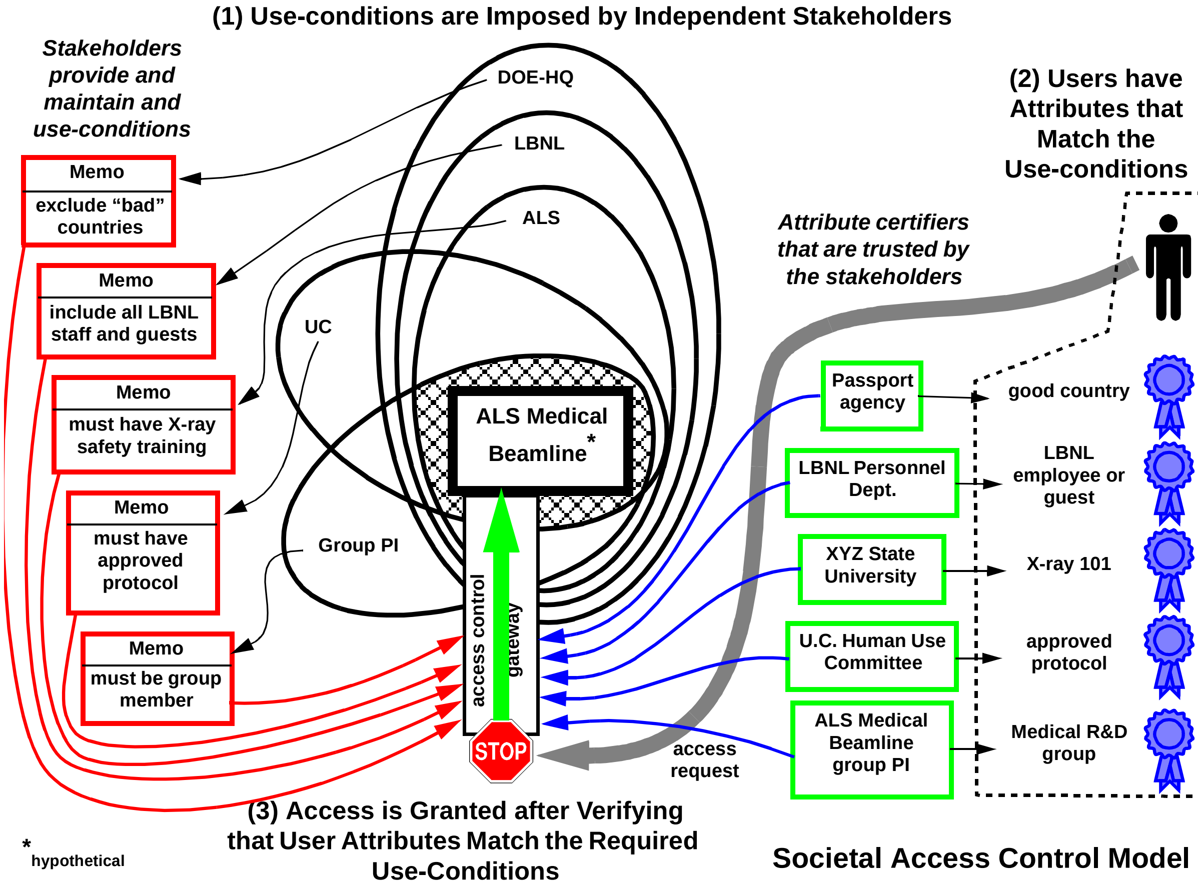
and these facilities, collaborations, and stakeholders are diffuse - geographically distributed and multi-organizational.

These circumstances require

- **distributed management of authorization- because the principals and resources are dispersed organizationally**
- **distributed access control - because the resources and users are dispersed geographically**

7.2.2 Use a Well Understood Approach as a Model

- ♦ **Stakeholders are identified by (usually) written policy**
- ♦ **Representations of authority (“use-conditions”) are made by written, signed procedures, memoranda, etc.**
- ♦ **The required use-conditions are satisfied by a set of attributes: organizational membership, training, etc.**



Societal Access Control Model

- Who and/or what can attest to users’ satisfaction of the use-conditions is established by policy: e.g. a token issued by a personnel department, a certificate of training issued by an accredited school, etc.
- Credential checking is usually based on an operational authority that compiles a list of stakeholder use-conditions and then validates the users’ attributes against this list

- **All of the attributes that match use-conditions are likely to be packaged into a "capability" - a single document (e.g. a "license" or badge) that names the user, and perhaps the resource and the range of permitted actions**
- **The access control enforcer - a door guard, the experiment PI, etc. - typically just validates the capability (e.g., checks the license) when access is requested**

This general societal model provides us with the framework for an on-line architecture that accomplishes the same sort of access control for on-line resources.

7.2.3 Overall Goals

On-line access control for the scientific environment must provide:

- **Secure sharing of resources in a way the reflects currently accepted practice and principles:**
  - **stakeholders independently make assertions about resource use**
  - **trusted third-parties certify user attributes required for the use-conditions**
  - **authenticated users that posses the required attributes easily gain access**

- the level of credential checking (and security) is determined by the nature of the resource being protected

♦ Dynamic and easily used mechanisms for generation, maintenance, and distribution of the access control information
  - those that make assertions (e.g. establish the use-conditions or attest to user attributes) must be able to do so within their own working environment (usability!)

♦ Strong assurances that use-conditions are met

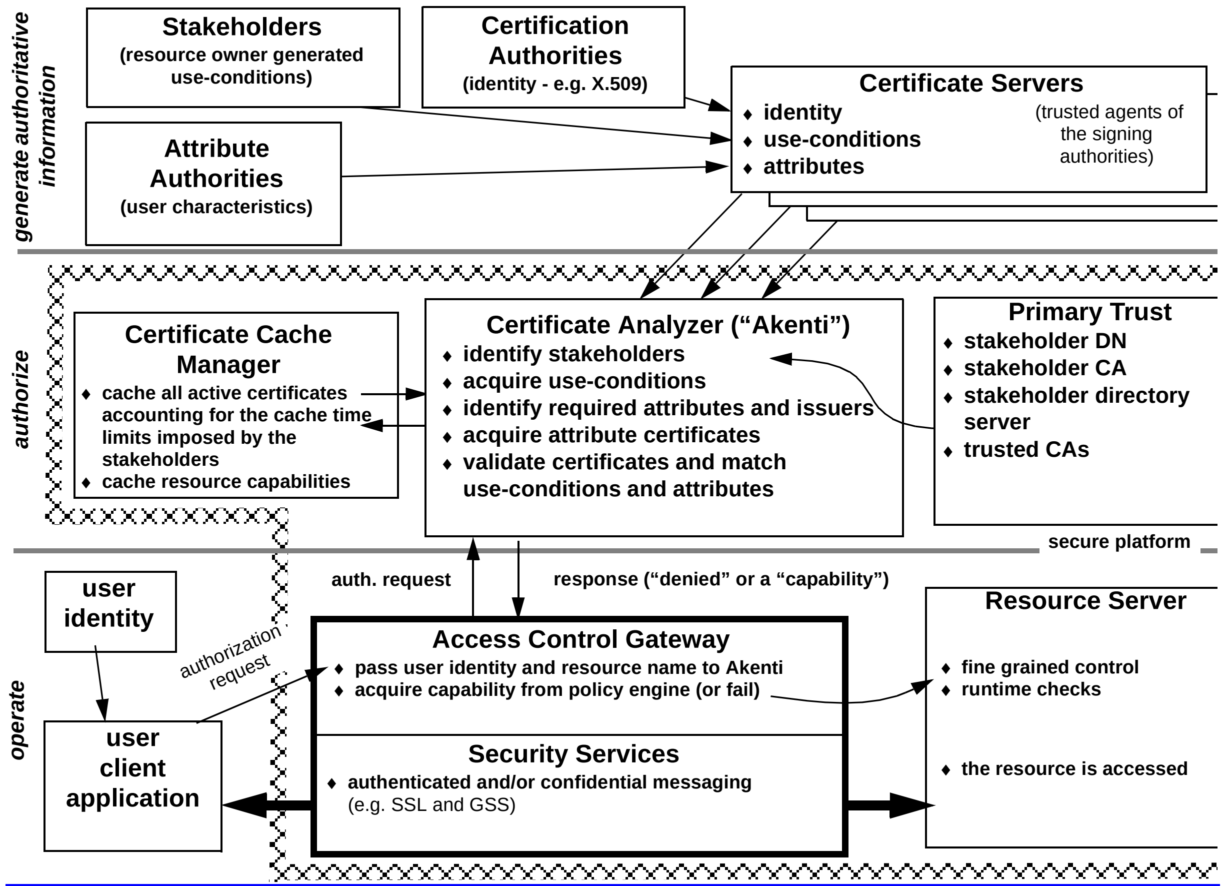
- access decisions must be made based on assured information and then enforced by strong security services

♦ A policy-neutral mechanism
  - policy is a reflection of agreements among humans
  - represent policy by assured “pointers” - digitally signed documents containing keywords and values (the type and name of the underlying policy agreement)
  - access control is by data driven certificate validation and analysis

7.2.4 General Approach

- ♦ Stakeholders are associated with resources by trusted third parties (currently in configuration files associated with the resource)
- ♦ All other "trust" is explicit
  - trusted CAs are represented by their names and public keys
  - certificate issuers are represented by their names and CAs
- ♦ Akenti is basically a data driven certificate analyzer
  - user identity and resource identity are presented
  - stakeholders are identified

  - use-conditions are collected and verified
  - required attributed are located and verified
  - result is packaged as a "capability" and passed to the resource access control gateway
- ♦ <u>Use-condition certificates</u>
  - allow stakeholders to impose their requirements in a "natural and convenient" way - by representing them as digitally signed documents that are generated, maintained, and distributed in the stakeholder's "local" (working) environment
  - use-conditions are expressed as named attributes having required values (usually "membership in named groups")



- may include "scope" - to support a hierarchical policy model - and lists of permitted actions (vis a vis a particular resource)
- can also be expressed as required values for components of DN

♦ Attribute certificates

- attribute certificates assign characteristics to DNs (i.e. named attribute has value X - frequently thought of as placing DN in named group)
- issued by trusted third parties (named in use-condition certificate) from their working environment, like use-condition certificates

- can carry auxiliary information that has meaning for resource access control gateway (e.g. permitted operations, objects to access, valid times of day, etc.)
- many namespace problems - in general, will probably have to establish an authoritative mapping function

♦ <u>Identity</u>

- standard X.509 certificates and Certification Authority infrastructure are used for identifying and authenticating various entities
- X.509 cert. acts as attribute certificate when requirement is DN component

Science Grid Vision

Science grids will provide uniformity, location independence, and capabilities for building complex, on-demand, large scale distributed systems from computing, data storage, instruments, and intellectual resources that are spread across the DOE Labs, NASA Centers and their partners.

This will lead to revolutionary new capabilities for solving large-scale science and engineering problems.

References and Acronyms

[1] Globus is a middleware system that provides a suite of services designed to support high performance, distributed applications. Globus provides:

- Resource Management: Components that provide standardized interfaces to various local resource management systems (GRAM) manage allocation of collections of resources (DUROC). All Globus resource management tools are tied together by a uniform resource specification language (RSL).
- Remote Access: Components that enable remote access to files (GASS and RIO) and executables (GEM).
- Security: Support for single sign-on, authentication, and authorization within the Globus system (GSI) and (experimentally) authorization (GAA).
- Fault Detection: Basic support for building fault detection and recovery into Globus applications.
- Information Infrastructure: Global access to information about the state and configuration of system components of an application (MDS).
- Grid programming services: Support writing parallel-distributed programs (MPICH-G), monitoring (HBM), etc.

www.globus.org provides full information about the Globus system.

[2] *The Grid: Blueprint for a New Computing Infrastructure*, edited by Ian Foster and Carl Kesselman. Morgan Kaufmann, Pub. August 1998. ISBN 1-55860-475-8. http://www.mkp.com/books_catalog/1-55860-475-8.asp

[3] "Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid," William E. Johnston, Dennis Gannon, and Bill Nitzberg. Eighth IEEE International Symposium on High Performance Distributed Computing, Aug. 3-6, 1999, Redondo Beach, California. (Available at http://www.nas.nasa.gov/~wej/IPG)

[4] "Vision and Strategy for a DOE Science Grid" - http://www.itg.lbl.gov/~wej/Grids

[5] See www.nas.nasa.gov/IPG for project information and pointers.

[6] Numerical Aerospace Simulation Systems Division ("NAS") of NASA Ames Research Center. www.nas.nasa.gov

[7] NASA's Information Technology (IT) Research and Technology (R&T) Base Program pioneers the identification, development, verification, transfer, and application of high-payoff aerospace technologies. The ACNS project responds to the requirements of the Information Technology Program and Aerospace Technology Enterprise by investing in simulation-based approaches to aerospace vehicle design, manufacture, and operations. See http://www.nas.nasa.gov/IT

[8] NASA's Consolidated Supercomputing Management Office (CoSMO) serves the "production" computing needs of the National Aeronautics and Space Administration. http://www.nas.nasa.gov/ACSF/

[9] See http://www-itg.lbl.gov/NGI/ for project information and pointers.

[10] A collaborative effort to enable desktop access to remote resources including, supercomputers, network of workstations, smart instruments, data resources, and more - computingportals.org

[11] The Grid Forum (www.gridforum.org) is an informal consortium of institutions and individuals working on wide area computing and computational Grids. Current working groups include Security (authentication, authorization), Scheduling and Resource Management, Grid Information Services, Application and Tool Requirements, Advanced Programming Models, Grid User Services and Operations, Account Management, Remote Data Access, Grid Performance

[12] "New Capabilities in the HENP Grand Challenge Storage Access System and its Application at RHIC" http://rncus1.lbl.gov/GC/docs/chep292lp1.doc
"STACS is ... responsible for determining, for each query request, which events and files need to be accessed, to determine the order of files to be cached dynamically so as to maximize their sharing by queries, to request the caching of files from HPSS in tape optimized order, and to determine dynamically which files to keep in the disk cache to maximize file usage."

[13] "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets." A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, (to be published in the Journal of Network and Computer Applications).

[14] "Storage Access Coordination Using CORBA," A. Sim, H. Nordberg, L.M. Bernardo, A. Shoshani and D. Rotem. Proceedings of the International Symposium on Distributed Objects and Applications. See http://gizmo.lbl.gov/sm/

[15] **Akenti**: "**Certificate-based Access Control for Widely Distributed Resources**," Mary Thompson, William Johnston, Srilekha Mudumbai, Gary Hoo, Keith Jackson, Usenix Security Symposium '99. Mar. 16, 1999. (See http://www-itg.lbl.gov/Akenti)

[16] GAA: "**Generic Authorization and Access control API**" (GAA API). IETF Draft. http://ghost.isi.edu/info/gss_api.html)

[17] Storage Resource Broker (SRB) provides uniform access mechanism to diverse and distributed data sources. http://www.sdsc.edu/MDAS/

[18] Condor is a High Throughput Computing environment that can manage very large collections of distributively owned workstations. http://www.cs.wisc.edu/condor/

[19] SCIRun is a scientific programming environment that allows the interactive construction, debugging and steering of large-scale scientific computations. http://www.cs.utah.edu/~sci/software/

[20] Ecce - www.emsl.pnl.gov

[21] WebFlow - A prototype visual graph based dataflow environment, WebFlow, uses the mesh of Java Web Servers as a control and coordination middleware, WebVM. See http://iwt.npac.syr.edu/projects/webflow/index.htm

[22] "QoS as Middleware: Bandwidth Reservation System Design." Gary Hoo and William Johnston, Lawrence Berkeley National Laboratory, Ian Foster and Alain Roy, Argonne National Laboratory and University of Chicago. To appear, Eighth IEEE International Symposium on High Performance Distributed Computing, Aug. 3-6, 1999, Redondo Beach, California. (See http://www-itg.lbl.gov/Clipper/QoS)

[23] "Numerical Propulsion System Simulation (NPSS) is a concerted effort by NASA Glenn Research Center, the aerospace industry and academia to develop an advanced engineering environment - or integrated collection of software programs - for the analysis and design of aircraft engines and, eventually, space transportation components. Its purpose is to dramatically reduce the time, effort and expense necessary to design and test jet engines. It accomplishes that by generating sophisticated computer simulations of an aerospace object or system, thus permitting an engineer to "test" various design options without having to conduct costly and time-consuming real-life tests. The ultimate goal of NPSS is to create a "numerical test cell" that enables engineers to create complete engine simulations overnight on cost-effective computing platforms." See http://hpcc.grc.nasa.gov/hpcc2/npssintro.shtml